IN THE CLAIMS:

Please amend the claims as listed below. This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method comprising:

determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with [[a]] said second data block, wherein a merged data block is formed.

2. (Original) The method of claim 1 further comprising receiving said first data block and said second data block from a bitstream.

3. (Original) The method of claim 1 wherein said first data block and said second data block are loaded from bitstream data in memory.

4. (Original) The method of claim 2 further comprising extracting a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

5. (Original) The method of claim 4 wherein said merged variable length symbol is sent to a register.

6. (Currently Amended) The method of claim 4 further comprising:

evaluating said merged data block for variable length symbols;

extracting any whole variable length symbols located in said second data block; and

sending ~~an~~ any extracted whole variable length symbols to a register.

7. (Original) The method of claim 6 wherein each of said variable length symbols is comprised of at least N-bits of data.

8. (Original) The method of claim 7 wherein N is a number greater than one.

9. (Original) The method of claim 2 wherein said first data block is held in a first single instruction multiple data register.

10. (Currently Amended) The method of claim [[3]] 9 wherein said first single instruction multiple data register is a buffer.

11. (Original) The method of claim 2 wherein said second data block is held in a second single instruction multiple data register.

12. (Currently Amended) The method of claim 1 wherein said shift merge operation

is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on [[a]] the specified shift count.

13. (Original) The method of claim 10 wherein said shift merge operation operates on data elements at a byte granularity.

14. (Original) The method of clam 10 wherein said shift merge operation operates on data elements at a bit granularity.

15. (Currently Amended) An apparatus comprising:

an execution unit to execute a plurality of instructions for a variable length decoding algorithm, wherein one of said instructions is a first instruction for a shift merge operation, said plurality of instructions to cause said execution unit to:

determine whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

perform a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with [[a]] said second data block, wherein a merged data block is formed.

16. (Original) The apparatus of claim 15 wherein plurality of instructions further cause said execution unit to extract a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

17. (Original) The apparatus of claim 15 wherein said first data block and said second data block are received from a bitstream.

18. (Original) The apparatus of claim 16 wherein said plurality of instructions further cause said execution unit to: evaluate said merged data block for variable length symbols; and extract any whole variable length symbols located in said second data block.

19. (Original) The apparatus of claim 18 wherein each of said variable length symbols is comprised of at least two bits of data.

20. (Original) The apparatus of claim 19 wherein said first data block and said second data block are held in a first single instruction multiple data register and a second single instruction multiple data register, respectively.

21. (Original) The apparatus of claim 20 wherein said first single instruction multiple data register is a buffer.

22. (Currently Amended) The apparatus of claim 21 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on [[a]] the specified shift count.

23. (Original) The apparatus of claim 22 wherein said shift merge operation is to operate on data elements at a byte granularity.

24. (Currently Amended) An article comprising a tangible machine readable medium that stores a program, said program being executable by a machine to perform a method comprising:

determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with [[a]] said second data block, wherein a merged data block is formed.

25. (Original) The article of claim 24 wherein said first data block and said second data block are to be loaded from a data bitstream.

26. (Original) The article of claim 25 wherein said method further comprises extracting a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

27. (Original) The article of claim 26 wherein each of said variable length symbols is comprised of at least two bits of data.

28. (Currently Amended) The article of claim 24 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge

of data operands based on [[a]] the specified shift count.

29. (Original) The article of claim 26 wherein said method further comprises: evaluating said merged data block for variable length symbols; and extracting any whole variable length symbols located in said second data block.

30. (Currently Amended) A system comprising: a memory to store data and instructions, a processor coupled to said memory on a bus, said processor operable to perform instructions for a variable length decoding algorithm, said processor comprising:

a bus unit to receive a sequence of instructions from said memory;

an execution unit coupled to said bus unit, said execution unit to execute said sequence, said sequence to include a first instruction specifying a first data block, a second data block and a shift count for a shift merge operation, said sequence to cause said execution unit to:

determine whether any unprocessed data bits for a partial variable length symbol exist in [[a]] the first data block; and

perform [[a]] the shift merge operation responsive to the first instruction to merge said unprocessed data bits from said first data block with [[a]] said second data block, wherein a merged data block is formed.

31. (Original) The system of claim 30 wherein said first data block and said second data block are loaded from bitstream data.

32. (Original) The system of claim 30 wherein said plurality of instructions further cause

said execution unit to extract a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

33. (Original) The system of claim 32 wherein said plurality of instructions further cause said execution unit to: evaluate said merged data block for variable length symbols; and extract any whole variable length symbols located in said second data block.

34. (Original) The system of claim 33 wherein each of said variable length symbols is comprised of at least two bits of data.

35. (Original) The system of claim 34 wherein said first data block and said second data block are held in a first single instruction multiple data register and a second single instruction multiple data register, respectively.

36. (Currently Amended) The system of claim 35 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on [[a]] the specified shift count.

37. (Original) The system of claim 36 wherein at least a single edge for at least one or more of said variable length symbols is not aligned at a byte boundary.

38. (Original) The system of claim 37 wherein said shift merge operation is to operate on data elements at a bit granularity.

39. (Original) The system of claim 37 wherein said shift merge operation is to operate on data elements at a byte granularity.